



Représentation par objets et classification pour le raisonnement à partir de cas

Jean Lieber, Amedeo Napoli

► To cite this version:

Jean Lieber, Amedeo Napoli. Représentation par objets et classification pour le raisonnement à partir de cas. Conférence AFCET - RFIA, Université de Clermont-Ferrand, 1998, Clermont-Ferrand, France. pp.345-354 (tome III). inria-00098720

HAL Id: inria-00098720

<https://inria.hal.science/inria-00098720>

Submitted on 26 Sep 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Représentations par objets et classification pour le raisonnement à partir de cas

Object-Based Representation and Classification for Case-Based Reasoning

Jean Lieber et Amedeo Napoli
CRIN CNRS – INRIA Lorraine,
BP 239 – 54506 Vandœuvre-lès-Nancy Cedex
(Email: {lieber,napoli}@loria.fr)

Résumé

Dans cet article, nous proposons une analyse du raisonnement à partir de cas — RÀPC — dans le cadre des représentations de connaissances par objets. Nous montrons les rapports existant entre le RÀPC et le processus de classification, et nous mettons en valeur les classifications dure et élastique, qui interviennent lors de l'opération de remémoration dans le cycle du RÀPC, ainsi que la notion de chemin de similarité, qui intervient lors de l'étape d'adaptation. L'article se termine par une discussion sur les apports du travail présenté ici et sur une comparaison avec des travaux similaires.

Mots Clef

Raisonnement à partir de cas, représentation de connaissances par objets, classification.

Abstract

In this paper, we propose an analysis of case-based reasoning –CBR– in the context of object-based representation systems. We show the links existing between CBR and the classification process, and we introduce the strong and smooth classification processes used for the retrieval operation of the CBR cycle, and similarity paths used for the adaptation operation. The article ends with a discussion on the contributions of the present work and a comparison with related works.

Keywords

Case-based reasoning, object-based representation systems, classification.

1 Introduction

Le raisonnement à partir de cas — abrégé en RÀPC — associe à la donnée d'un problème P une solution, qui est construite en réutilisant la solution mémorisée

d'un problème P' jugé similaire à P . Le cycle associé à ce mode de raisonnement se compose de trois opérations principales : la remémoration pendant laquelle est recherché, dans une base de cas, le problème similaire P' , l'adaptation pendant laquelle la solution du problème similaire P' est adaptée pour être réutilisée, et la mémorisation éventuelle du problème P et de sa solution, en vue d'une réutilisation future. L'implantation de ce mode de raisonnement a conduit à la notion de système de RÀPC, qui s'appuie sur le cycle précédent et qui exploite une base de cas, mais éventuellement aussi une autre base de connaissances, pour résoudre un problème, qu'il soit de conception, d'interprétation, de diagnostic, de planification, etc.

Dans cet article, nous présentons une analyse des liens existant entre les représentations de connaissances par objets — abrégés en RCO —, le raisonnement par classification et le raisonnement à partir de cas. Les RCO et le raisonnement par classification peuvent être utilisés avec profit pour concevoir des systèmes à bases de connaissances [Napoli et Laurengo, 1993], et s'associent de façon très naturelle au RÀPC dès lors que les cas sont représentés par des objets organisés en hiérarchie [Napoli, 1994]. L'analyse présentée ici a pour ambition d'appréhender plus rigoureusement le comportement d'un système de RÀPC dans le contexte des RCO, pour pouvoir comparer un tel système de RÀPC à d'autres systèmes de résolution de problèmes, comme les systèmes logiques par exemple. Elle devrait permettre aussi de comparer les différentes sortes de systèmes de RÀPC entre eux, afin de faire émerger les points communs et les différences, mais aussi de dégager une méthodologie de conception de systèmes de RÀPC.

Un certain nombre de travaux sur la formalisation du raisonnement à partir de cas ont déjà été réalisés,

et notre principale source d'inspiration dans la suite est [Koehler, 1996], où le RÀPC, plus précisément la planification à partir de cas, est décrit dans un cadre logique : les cas sont représentés par des formules logiques et sont organisés *via* un processus d'indexation dans une hiérarchie d'index de plans, index qui sont eux-mêmes représentés par des concepts dans une logique de descriptions (ou logique terminologique [Nebel, 1990]). Le rapport entre les deux formalismes de représentation est analysé et discuté, en montrant ce qu'un tel processus d'indexation apporte au processus de RÀPC, mais aussi quelles propriétés doivent vérifier les deux formalismes de représentation pour que leur adéquation soit correcte et complète. Dans cet article, nous reprenons et étendons ces travaux dans le cadre d'une RCO, en distinguant différentes notions de classification — les classifications dure et élastique — et en introduisant la notion de chemin de similarité, qui garantit qu'un cas analogue, une fois mémoré, peut être adapté pour résoudre le problème courant. La représentation des problèmes, des index et des relations entre problèmes et index revêt une grande importance. Ainsi, l'organisation hiérarchique des index dans une RCO favorise la recherche de cas analogues. De plus, les relations entre problèmes et index doivent être significatives, au sens où elles permettent d'aboutir à une solution pour le problème courant.

L'article se compose de quatre paragraphes principaux. Dans le deuxième paragraphe sont rappelées quelques généralités nécessaires sur le RÀPC et notre façon de voir l'indexation. Dans le troisième paragraphe, le RÀPC est présenté et mis en valeur dans le contexte d'une RCO : une relation de généralité entre problèmes, ainsi que les opérations de remémoration et d'adaptation, et leur lien avec la classification, y sont détaillés. Enfin, dans le quatrième paragraphe, l'approche présentée ici est discutée et les liens avec des travaux similaires sont explicités, avant de conclure.

2 Le RàPC et l'indexation

Le raisonnement à partir de cas peut se voir comme une fonction RàPC, qui, à l'énoncé d'un nouveau problème ou problème *cible* P , fait correspondre une solution $Sol(P)$, en tirant parti d'un ensemble de *cas*, qui sont des problèmes déjà résolus accompagnés de leur solution. Un cas mémorisé, ou cas *source*, est la donnée d'un couple problème – solution $(P, Sol(P))$, et une *base de cas* est un ensemble fini de cas $Base_de_cas = \{(P_k, Sol(P_k)) / k = 1, \dots, q\}$. Il est classique de considérer le problème cible comme un nouveau cas *cas-cible* $= (cible, \perp)$ — où \perp correspond à l'absence de solution — pour lequel il faut construire $Sol(cible)$ en s'aidant d'un cas mémorisé *cas-source* $= (source,$

$Sol(source))$.

La fonction RàPC se décompose en trois opérations principales, la *remémoration*, l'*adaptation* et la *mémorisation*. Étant donné un problème *cible* à résoudre, la remémoration consiste à retrouver dans la base de cas un problème source *source*, jugé *similaire* — ou *analogue* — à *cible*. L'adaptation consiste à adapter $Sol(source)$ pour produire une solution de *cible*. Une étape de mémorisation peut venir compléter les deux étapes précédentes, où, selon l'intérêt associé au problème *cible* et à sa solution, le couple $(cible, Sol(cible))$ peut être mémorisé dans la base de cas, accompagné le cas échéant d'informations liées à la construction de la solution $Sol(cible)$ lors des opérations de remémoration et d'adaptation.

Une base de cas peut éventuellement être munie d'une relation d'ordre, qui permet de comparer les cas entre eux et de déterminer leur *degré de généralité*. Toutefois, ce n'est généralement pas sur les cas eux-mêmes, ni sur l'énoncé des problèmes qu'est définie une telle relation d'ordre, mais plutôt sur une *généralisation* de cet énoncé, appelée *index* : si $K = (source, Sol(source))$, alors l'index de K , qui est noté $idx(source)$, est un problème « plus général que » *source* (le sens exact de « plus général que » est explicité au paragraphe 3.2). L'index d'un cas est censé décrire les caractéristiques essentielles qui sont associées à l'énoncé du problème et à la construction de sa solution.

Il faut faire deux remarques propres à l'approche présentée ici.

[R₁] La première est qu'un index $idx(source)$ est un problème au même titre que l'énoncé *source* lui-même. En particulier, les énoncés de problèmes et les index sont supposés être tous représentés dans le même formalisme, une RCO, et sont donc comparables.

[R₂] La seconde est que l'indexation ne se conçoit qu'en présence d'une solution : l'index d'un cas $K = (source, Sol(source))$ vérifie le fait que la solution $Sol(source)$ est généralisable en une solution de $idx(source)$. En particulier, pour un problème cible $(cible, \perp)$, $idx(cible)$ est identique à *cible* : puisque $Sol(cible)$ n'est pas connue, l'ensemble de *cible* est appelé à jouer un rôle dans le processus de résolution du problème cible (autrement dit, l'index de *cible* est *cible* elle-même).

Intuitivement, $idx(source)$ matérialise la partie de *source* qui joue directement un rôle dans la solution $Sol(source)$, ce qui s'accorde avec la notion d'*empreinte* — *footprints* — telle qu'elle est introduite dans [Veloso, 1994]. L'indexation de $K = (source, Sol(source))$ permet d'obtenir $idx(source)$ et consiste à enlever dans *source* les éléments qui ne jouent pas de rôle

dans $\text{Sol}(\text{source})$. Lorsqu'on dispose d'un processus qui permet de décider si un élément de source joue ou ne joue pas de rôle dans $\text{Sol}(\text{source})$, alors le processus d'indexation peut être automatique — comme dans [Velo, 1994] et [Koehler, 1996] — mais il reste bien souvent manuel et dépendant de connaissances propres au domaine d'étude, voire de spécialistes, comme dans [Lieber, 1997].

3 Le RàPC dans le contexte d'une RCO

Dans ce paragraphe, nous présentons tout d'abord les représentations de connaissances par objets et le processus de classification. Puis, nous montrons comment certains problèmes de recherche peuvent être appréhendés dans l'univers hiérarchique des RCO, et sur cette base, comment les opérations de remémoration et d'adaptation sont effectuées pour résoudre un nouveau problème, en tirant avantageusement parti du processus de classification.

3.1 Éléments sur les RCO

Une RCO s'appuie sur une hiérarchie $\mathcal{H} = (\mathcal{X}, \sqsubseteq, \omega)$, qui est un graphe orienté sans circuit, où \mathcal{X} est un ensemble de classes (les sommets du graphe), \sqsubseteq est une relation d'ordre partiel, ou relation de *subsumption*, et ω est l'élément maximal de \mathcal{X} suivant \sqsubseteq ; ω est appelée la *racine* de la hiérarchie et elle est supposée toujours exister. Une classe C *spécialise* une classe D — ou encore C est *subsumée* par D — ce qui se note $C \sqsubseteq D$, et la donnée de cette relation constitue un arc de la hiérarchie.

Une classe C représente un concept du monde réel et possède une identité et un ensemble de propriétés correspondant aux caractéristiques du concept représenté. Du point de vue de l'implantation, une classe est une entité générique qui possède un nom et qui définit un modèle à partir duquel s'engendrent ses instances, qui représentent des entités individuelles. Dans ce qui suit, une classe C se décrit par la conjonction : $C = (a_1, s_1) \sqcap (a_2, s_2) \sqcap \dots \sqcap (a_n, s_n)$, où les a_k sont des attributs et les s_k une ou plusieurs spécifications attachées à l'attribut et précisant le type, le domaine et la cardinalité des valeurs de l'attribut (les a_k sont différents deux à deux). Une classe C spécialise une classe D par adjonction de nouvelles propriétés et/ou par complétion de propriétés déjà définies. Pour déterminer que D subsume C , il faut vérifier que, pour tout attribut a_k dans D , il existe un attribut a_k — donc de même nom — dans C , tel que les spécifications associées à a_k dans D soient vérifiées par a_k dans C (les spécifications attachées à un attribut et les rapports existant entre spécifications ne sont pas détaillés dans le cadre de cet

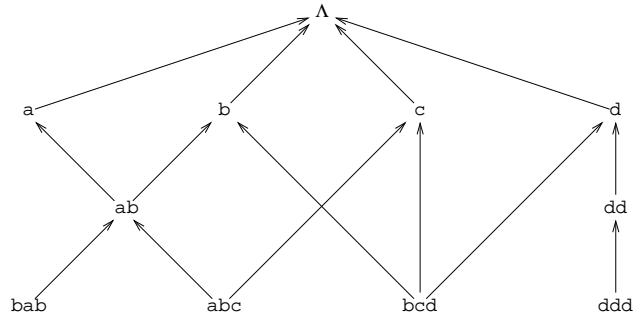


FIG. 1 — La hiérarchie \mathcal{A} des mots de l'ensemble $\{\Lambda, a, ab, abc, b, bab, bcd, c, d, dd, ddd\}$; Λ , qui désigne le mot vide, est la racine de la hiérarchie \mathcal{A} .

article, voir par exemple [Napoli, 1995] et [Chouvet *et al.*, 1996]). Enfin, une instance i de la classe C se note $i = (a_1, v_1) \sqcap (a_2, v_2) \sqcap \dots \sqcap (a_n, v_n)$, où v_k est la valeur associée à a_k pour l'instance i (avec éventuellement $v_k = \emptyset$).

La figure 1 montre un exemple simple de hiérarchie de mots, notée \mathcal{A} . Une classe détermine un ensemble de mots qui contiennent chacun un motif défini sur l'alphabet $\{a, b, c, d\}$. Le nom de la classe est donné par le motif qui la caractérise : ainsi, $ab = (\text{motif}, *ab*)$, où *motif* est un nom d'attribut et $*$ dénote une chaîne quelconque. De plus, étant donné deux classes x et y , $x \sqsubseteq y$ si le motif y est un sous-motif du motif x , en d'autres termes, si x s'écrit $m \cdot y \cdot m'$, où m et m' sont deux mots.

Le processus de classification opère sur une hiérarchie $\mathcal{H} = (\mathcal{X}, \sqsubseteq, \omega)$ et cherche à mettre en évidence les dépendances implicites qui existent entre les objets dans \mathcal{H} , dépendances classes – classes et dépendances classes – instances. En particulier, le processus de classification permet de placer un objet x , classe ou instance, dans la hiérarchie \mathcal{H} .

Le *raisonnement par classification* s'appréhende comme une procédure de déduction opérant sur une hiérarchie [Napoli et Laurenço, 1993], dont la mise en œuvre repose sur un cycle comprenant trois étapes :

- (i) *instanciation* : création d'un nouvel objet x (qui peut être une classe ou une instance),
- (ii) *classification* : parcours de la hiérarchie \mathcal{H} , recherche des subsumants les plus spécifiques de x (SPS), recherche des subsumés les plus généraux de x (SPG) et mise en place de x dans \mathcal{H} ,
- (iii) *exploitation* : la mise en place de x dans \mathcal{H} déclenche des opérations de mise à jour d'objets interdépendants et/ou la production de nouveaux objets, ce qui ramène le cycle à sa première étape.

La figure 2 reprend l'exemple de la hiérarchie \mathcal{A} et montre le résultat de la classification et de l'insertion

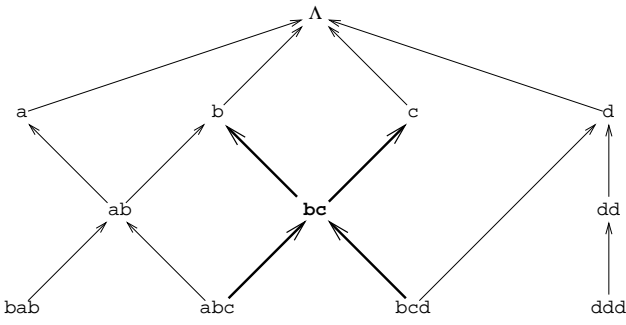


FIG. 2 — La classification et l'insertion du mot *bc* dans la hiérarchie *A*. Les SPS de *bc* sont *b* et *c*, et les SPG de *bc* sont *abc* et *bcd*. Les nouveaux arcs sont en gras ; par convention, les arcs de transitivité *abc* → *c* et *bcd* → *b* sont supprimés.

du mot *bc* dans la hiérarchie *A*.

Un exemple important d'utilisation du raisonnement par classification est le traitement des requêtes dans une hiérarchie \mathcal{H} [Borgida et McGuinness, 1996] [Chouvet *et al.*, 1996]. Une requête est représentée par une classe *Q*, et, satisfaire la requête consiste à classer *Q* dans la hiérarchie \mathcal{H} . Les réponses à la requête sont toutes les instances des subsumés de la classe *Q* — qui sont nécessairement des instances de *Q* — mais aussi certaines instances des SPS de *Q* reconnues comme des instances de la classe *Q* elle-même (c'est-à-dire vérifiant les spécifications données dans *Q*).

Satisfaire une requête dans une hiérarchie permet de résoudre un problème d'interrogation de base (de connaissances) hiérarchique. Un tel problème d'interrogation n'est généralement qu'un sous-problème élémentaire associé à un problème plus complexe, qui peut être résolu dans le cadre des RCO. C'est cette catégorie de problèmes qui nous intéresse et qui est mise en valeur dans le paragraphe suivant.

3.2 Une relation de généralité entre problèmes dans une RCO

Dans cet article, nous nous intéressons à une catégorie particulière de problèmes de recherche, dont l'énoncé ou l'index associé à l'énoncé peuvent se représenter par une classe (ou encore par l'instance d'une classe). Cette restriction est certes limitative, mais elle recouvre tout de même un nombre assez considérable de problèmes courants, comprenant des problèmes de planification [Koehler, 1996] [Lieber et Napoli, 1996], de filtrage [Chouvet *et al.*, 1996], d'interrogation de bases de données ou de connaissances [Borgida et McGuinness, 1996] et de gestion de graphes (conceptuels) [Mugnier et Chein, 1996]. Ainsi, un problème générique de filtrage comme « étant donné une hiérarchie

\mathcal{H} , trouver l'ensemble des instances de la classe *x* vérifiant la contrainte *S* » peut se combiner, par l'intermédiaire d'opérations booléennes (**et**, **ou** et **non**), avec d'autres énoncés de même type pour appréhender des problèmes de nature complexe. Dans le même ordre d'idées, il faut encore remarquer que, dans les problèmes de classification, de configuration et de diagnostic, l'organisation des divers éléments associés à la résolution d'un problème — connaissances, données, solutions, etc. — repose généralement sur une hiérarchie [Stefik, 1995].

Nous allons maintenant définir une relation de généralité entre problèmes, dans le cadre d'une RCO. Remarquons tout d'abord qu'une formule logique $\text{gen}(\varphi)$ est une *généralisation* de la formule logique φ si $\varphi \rightarrow \text{gen}(\varphi)$ est vrai. En particulier, *A* est une généralisation de la conjonction $A \wedge B$, car $A \wedge B \rightarrow A$, où *A* et *B* sont des variables propositionnelles.

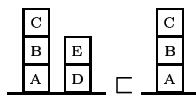
De la même façon, une spécification élémentaire *s* est *plus forte* ou *plus contraignante* qu'une spécification élémentaire *s'* si $s \rightarrow s'$. Ainsi, $s = x \geq 40 \rightarrow s' = x \geq 30$. Cette relation entre spécifications élémentaires est employée dans le contexte des RCO pour comparer les types et les domaines de valeurs (les domaines spécialisants sont inclus dans les domaines spécialisés [Chouvet *et al.*, 1996]). En reprenant le formalisme introduit au paragraphe précédent, un couple attribut – spécification (*a*, *s'*) est plus contraignant qu'un couple (*a*, *s*) si $s' \rightarrow s$.

La relation entre spécifications élémentaires nous permet de définir une relation de généralité entre énoncés de problèmes (qui est un préordre sur l'ensemble des problèmes considérés dans le cadre de cet article) : un énoncé de problème *P* est *plus général* qu'un énoncé de problème *P'* si, pour chaque couple attribut – spécification (*a*, *s*) dans *P*, il existe un couple attribut – spécification (*a*, *s'*) dans *P'* tel que $s' \rightarrow s$. En termes formels, il vient : $P = (a_1, s_1) \sqcap (a_2, s_2) \sqcap \dots \sqcap (a_n, s_n)$, est plus général que $P' = (b_1, s'_1) \sqcap (b_2, s'_2) \sqcap \dots \sqcap (b_m, s'_m)$ si pour tout couple (*a_k*, *s_k*) dans *P* il existe un couple (*b_ℓ*, *s'_ℓ*) dans *P'* tel que $a_k = b_\ell$ et $s'_\ell \rightarrow s_k$.

Par exemple, l'énoncé de problème *P* « trouver l'ensemble des personnes dont l'âge est supérieur ou égal à 30 », qui peut se représenter par la conjonction (*sorte-de* *Personne*) \sqcap (*âge* ≥ 30), est plus général que l'énoncé de problème *P'* « trouver l'ensemble des personnes dont l'âge est supérieur ou égal à 40 », qui peut se représenter par la conjonction (*sorte-de* *Personne*) \sqcap (*âge* ≥ 40), car $(\text{âge} \geq 40) \rightarrow (\text{âge} \geq 30)$. L'exemple que nous venons de voir est générique et possède de nombreuses variations comme « trouver un *x* ou l'ensemble des *x* ayant une propriété $y \leq z$ », « trouver un *x* ou l'ensemble des *x* qui possède la pro-

priété y », etc.

Nous donnons maintenant un autre exemple de généralité entre problèmes, dans le cadre du monde des blocs, et c'est cet exemple qui sera repris dans la suite.



Les entités du monde des blocs sont une table et des blocs étiquetés. Un bloc peut être posé sur la table ou sur un autre bloc. Le bloc A est *libre* si aucun bloc n'est posé sur lui, et c'est la condition qui doit être vérifiée pour qu'un bloc puisse être posé sur A. L'énoncé de gauche où sont voisines deux piles de blocs est moins général que l'énoncé de droite qui n'est constitué que d'une seule des deux piles de blocs, en l'occurrence C qui est posé sur B qui est posé sur A.

Un dernier exemple de généralité entre problèmes, qui se démarque des précédents, est présenté dans [Lieber et Napoli, 1996] et [Lieber, 1997]. Il s'agit d'un problème de planification en synthèse organique : construire une molécule M consiste à construire ses composants, et le problème de construire un composant est plus général que le problème de construire le composé, car la construction du composé passe par la construction du composant ; autrement dit, la contrainte associée à la construction du composant est moins forte que celle qui est associée à la construction du composé.

3.3 Remémoration, adaptation et classification

La procédure d'inférences associée au RÀPC se décompose en deux opérations principales, remémoration et adaptation. Nous allons voir dans ce qui suit comment peuvent s'appréhender ces deux opérations dans le contexte d'une RCO.

La remémoration. L'opération de remémoration consiste à retrouver un cas *cas-source* *similaire* au cas étudié *cas-cible*. La recherche du cas mémorisé ne se fait pas sur la donnée du cas cible, mais sur la donnée des index associés à *cas-cible* et *cas-source* : c'est grâce aux index qu'est évaluée la similarité entre *cas-cible* et *cas-source*. Contrairement à de nombreuses approches où la similarité s'appuie explicitement sur l'utilisation d'une mesure de similarité (voir par exemple [Rougegrez-Loriette, 1994] [Rifqi, 1996] [Mignot, 1997]), l'opération de remémoration est ici considérée comme une opération de classification de *cible* (rappelons que $\text{idx}(\text{cible}) = \text{cible}$, remarque [R₂] du paragraphe 2) dans la hiérarchie \mathcal{H}_{idx} . Plus précisément, l'opération de remémoration se divise en

deux étapes (comparables aux deux premières étapes du raisonnement par classification, voir § 3.1) :

- **Indexation** : construction de la représentation de *cible*.
- **Classification** : *cible* est classifiée dans \mathcal{H}_{idx} ; tous les problèmes sources dont l'index est un subsumant de *cible* fournissent une solution qui peut être potentiellement réutilisée pour produire une solution du problème cible *cible* (conformément à la relation de généralité entre problèmes).

La classification de *cible* dans \mathcal{H}_{idx} peut déterminer plusieurs cas sources, qui peuvent être réutilisés pour résoudre le problème cible. Si un ou plusieurs cas sources sont remémorés, alors un cas est sélectionné et l'opération d'adaptation est activée, ce qui correspond au processus de *classification dure*. En revanche, si aucun cas source n'est remémoré — le subsumant le plus spécifique de *cible* dans \mathcal{H}_{idx} est ω — alors le processus de *classification élastique* est activé.

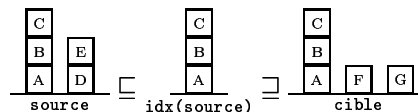
La classification dure, les chemins de similarité et l'adaptation. L'opération d'adaptation se combine avec l'opération de remémoration par l'intermédiaire de la notion de *chemin de similarité*, qui fait référence à une suite de relations liant l'énoncé du problème source *source* à l'énoncé du problème cible *cible*.

Plus précisément, le rôle du processus de classification dure est de retrouver des chemins de similarité de la forme :

$$\text{source} \sqsubseteq \text{idx}(\text{source}) \sqsupseteq \text{cible} \quad [1]$$

La relation $\text{source} \sqsubseteq \text{idx}(\text{source})$ indique que l'index du problème *source* est plus général que l'énoncé *source*, ce qui est le propre d'un index. La relation $\text{idx}(\text{source}) \sqsupseteq \text{cible}$ indique que l'index du problème *source* est plus général que l'index du problème cible, qui est *cible* elle-même.

Un exemple de chemin de similarité obtenu par classification dure dans le monde des blocs est le suivant :



Lorsqu'un chemin de similarité de la forme [1] est retrouvé, l'opération d'adaptation s'appuie essentiellement sur deux étapes, la *généralisation* et la *spécialisation*. Une première étape dans l'adaptation consiste à généraliser la solution $\text{Sol}(\text{source})$ associée au cas source pour produire une solution généralisée $\text{Sol}(\text{idx}(\text{source}))$ (c'est encore une fois l'objet de la

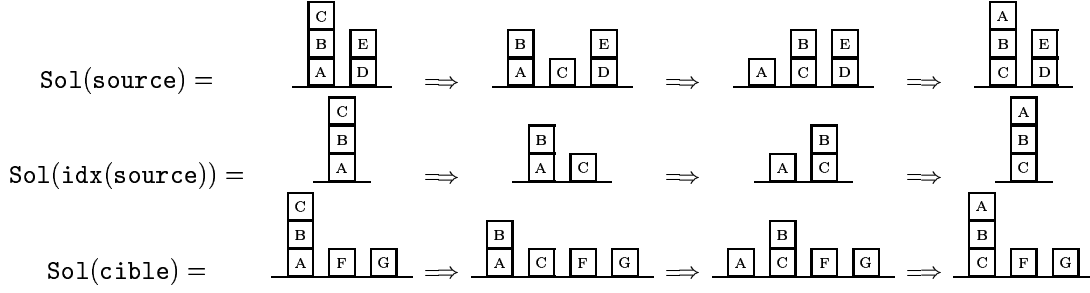


FIG. 3 – Un exemple d'adaptation à l'aide d'un chemin de similarité reposant sur une classification dure.

remarque [R₂] du paragraphe 2). Une seconde étape consiste à spécialiser $\text{Sol}(\text{idx}(\text{source}))$ en $\text{Sol}(\text{cible})$.

Un exemple d'adaptation dans le monde des blocs est donné à la figure 3. Cet exemple va nous servir aussi à expliciter la relation qui existe entre le problème *source* et l'index $\text{idx}(\text{source})$. Chaque ligne du schéma correspond à la résolution d'un problème avec ses différentes étapes. La ligne du haut montre la solution associée au cas *source*, qui est généralisée en la ligne du milieu, qui correspond au problème associé à l'index du cas *source*. Enfin, le problème cible correspond à la ligne du bas et la solution obtenue pour le cas cible est une spécialisation de la ligne du milieu. L'énoncé *source* comprend deux piles de blocs, CBA et ED de haut en bas ; l'index de cet énoncé est la seule pile CBA . En effet, le problème posé ici consiste à transformer la pile CBA en une pile ABC , sans que la pile ED ne soit modifiée. Ainsi, l'index $\text{idx}(\text{source})$ n'est concerné que par la seule pile CBA , et le processus de résolution du problème $\text{idx}(\text{source})$ ne comporte que les éléments nécessaires à la résolution du problème *source* : l'empreinte ici est constituée de la pile CBA à laquelle est associée son processus de transformation.

Le schéma de la figure 3 montre bien les deux étapes de généralisation – spécialisation permettant de trouver la solution d'un problème cible après classification dure. Ainsi, dès lors que le processus de classification dure a établi un chemin de similarité entre un énoncé *source* auquel est associée une solution $\text{Sol}(\text{source})$ et un énoncé *cible*, une suite d'opérations de généralisation et de spécialisation permet d'adapter la solution $\text{Sol}(\text{source})$ en une solution $\text{Sol}(\text{cible})$.

La classification élastique. Le processus de classification dure s'appuie sur un appariement *exact*. Cependant, il arrive que les caractéristiques des énoncés des problèmes associés aux cas de la base ne s'apparient pas directement avec les caractéristiques de l'énoncé du problème cible. En particulier, l'équation [1] n'est plus vérifiée, et le seul subsumant de *cible*

dans \mathcal{H}_{idx} est la classe ω . C'est alors qu'intervient le processus de *classification élastique* dont le principe est le suivant : « déformer » *source* et *cible* de telle sorte que l'équation [1] soit vérifiée par les énoncés déformés de *source* et *cible*. En d'autres termes, classification élastique = classification dure + déformations. Plus précisément, étant donné un ensemble de fonctions de déformation, le processus de classification élastique cherche à construire un chemin de similarité de la forme [2] :

$$\text{source} \sqsubseteq \text{idx}(\text{source}) \simeq \varphi(\text{idx}(\text{source})) \sqsupseteq \psi(\text{cible}) \simeq \text{cible}.$$

Les fonctions de déformation (élémentaires) dont il est question ici sont des fonctions de généralisation, de spécialisation, de changement de type et de modifications qui dépendent du domaine où le problème est posé. Dans l'équation [2], φ et ψ sont des fonctions de déformation qui sont supposées vérifier les propriétés suivantes :

- (i) si une solution $\text{Sol}(\text{idx}(\text{source}))$ est connue, alors une solution $\text{Sol}(\varphi(\text{idx}(\text{source})))$ peut être construite,
- (ii) si une solution $\text{Sol}(\psi(\text{cible}))$ est connue, alors une solution $\text{Sol}(\text{cible})$ peut être construite.

Les propriétés (i) et (ii) donnent le sens de la relation \simeq dans l'équation [2] : il est possible de passer de l'énoncé du problème $\text{idx}(\text{source})$ à l'énoncé $\varphi(\text{idx}(\text{source}))$ en conservant le fait que le problème considéré possède une solution ; il en va de même pour $\psi(\text{cible})$ et *cible*. Les propriétés (i) et (ii) sont importantes et jouent un rôle lors de l'opération d'adaptation (du même ordre que la généralisation d'une solution de *source* à $\text{idx}(\text{source})$), comme indiqué dans la remarque [R₂] du paragraphe 2). Ainsi, étant donné un chemin de similarité de la forme [2], l'opération d'adaptation s'appuie sur trois opérations, la généralisation et la spécialisation, comme lors de l'adaptation d'un chemin de similarité de la forme [1], avec en plus la prise en compte des déformations régies par les fonc-

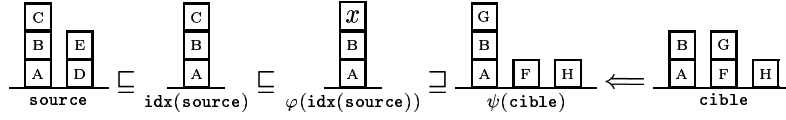


FIG. 4 — Un exemple de chemin de similarité obtenu après classification élastique. Le bloc étiqueté par x correspond au bloc générique, qui peut être substitué par tout autre bloc.

tions φ et ψ . L'adaptabilité des étapes de déformation repose alors sur les propriétés (i) et (ii).

Un exemple d'adaptation pour la classification élastique est complètement détaillé dans [Lieber, 1997]. Un autre exemple est donné à la figure 4, qui montre un chemin de similarité de la forme [2], obtenu après classification élastique, dans le monde des blocs. Dans cet exemple, l'application de la fonction φ consiste à passer du bloc étiqueté par C à un bloc générique étiqueté par la variable x : l'application de φ correspond donc à une généralisation d'étiquette ; x est supposée être une étiquette variable et le bloc dénoté par x peut être substitué par tout autre bloc. L'application de la fonction ψ consiste à déplacer le bloc G sur le bloc B, et correspond à un déplacement élémentaire de bloc.

La recherche de fonctions de déformation et le coût d'un chemin de similarité. Pour terminer, nous allons donner quelques explications sur la façon dont sont retrouvées et construites les fonctions φ et ψ . Schématiquement, l'équation [2] associée à la classification élastique passe par la mise en évidence de deux fonctions de déformation φ et ψ , telles que $\varphi(A) \sqsupseteq \psi(B)$, pour $A = \varphi(\text{idx}(\text{source}))$ et $B = \text{cible}$. En réalité, ces fonctions correspondent à la composition d'une suite d'applications de fonctions élémentaires de déformation : après une série d'applications de telles fonctions élémentaires, A est transformé en $\varphi(A)$ et B en $\psi(B)$. De plus, la contrainte suivante doit être vérifiée :

$$A \longrightarrow \dots \longrightarrow \varphi(A) \sqsupseteq \psi(B) \longleftarrow \dots \longleftarrow B.$$

Dans le monde des blocs par exemple, ces fonctions élémentaires de déformation sont la généralisation et le déplacement d'un bloc pour φ , le déplacement d'un bloc pour ψ . Pour mettre en évidence les compositions de fonctions élémentaires qui constituent les fonctions φ et ψ , une exploration « bidirectionnelle » dans un espace d'états particulier est effectuée, où :

- un état est donné par un couple de problèmes (A, B) ;
- un état successeur d'un état (A, B) est de la forme (A', B) ou de la forme (A, B') , où $A \longrightarrow A'$ et $B' \longleftarrow B$.

B : A' (respectivement B') est obtenu par application d'une fonction élémentaire sur A (respectivement sur B) ;

- un état (A, B) est un état but si $A \sqsupseteq B$.

La figure 5 représente une telle exploration bidirectionnelle. L'algorithme d'exploration utilisé dans notre cadre est un algorithme A^* , qui exploite une fonction d'évaluation associant un coût à un état, en comptabilisant les coûts liés aux déformations élémentaires $A \longrightarrow A'$ ou $B' \longleftarrow B$. Le coût d'un chemin entre A et B est la somme des coûts de déformations élémentaires constituant le chemin. Le coût d'une déformation élémentaire peut être fixé arbitrairement, mais peut aussi dépendre du domaine, comme dans le cas de l'application présentée dans [Lieber, 1997]. Le choix du coût d'une déformation élémentaire n'est pas indifférent, mais reste encore, pour notre part, à appréhender plus précisément.

Les difficultés de l'adaptation. Le processus de remémoration s'appuyant sur les classifications dure et élastique peut être qualifié de *remémoration guidée par l'adaptabilité*, comme présenté dans [Smyth et Keane, 1994] et [Smyth, 1996]. Il met en valeur la similarité entre deux problèmes, mais aussi « en quoi » les deux problèmes sont similaires, et comment « passer » de l'un à l'autre par l'intermédiaire d'un chemin de similarité. Le meilleur cas source devant être adapté pour résoudre le problème cible est alors celui qui réalise le chemin de similarité de coût minimal, un coût élémentaire étant associé à chaque opération du chemin de similarité (généralisation, spécialisation et déformation).

Il faut noter que l'*adaptabilité* est une notion purement *syntactique* — qui ne dépend que de la capacité du système à proposer une solution par adaptation — s'opposant à la *validité sémantique* de l'adaptation, qui est sa capacité à fournir une solution correcte. Ainsi, le problème de la validité de la solution obtenue par adaptation se pose dans les mêmes termes que pour le raisonnement par analogie [Carbonell, 1983]. Pour notre part, nous considérons qu'il ne peut être réglé qu'en interaction avec l'extérieur, et nous n'aborderons pas le sujet plus avant dans le cadre de cet

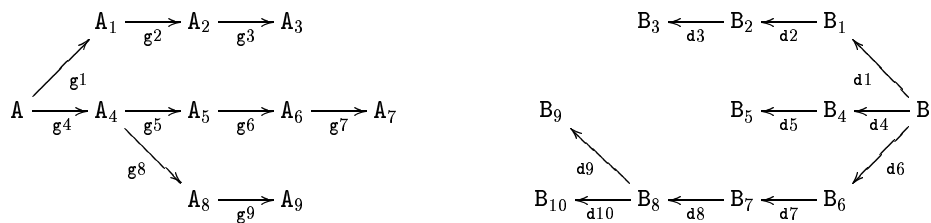


FIG. 5 — Une exploration A^* bidirectionnelle qui schématise la recherche des fonctions de déformations φ et ψ telles que $\varphi(A) \sqsupseteq \psi(B)$. La recherche de φ s'effectue à gauche — côté source — et φ résulte d'une composition de fonctions élémentaires g_i . La recherche de ψ s'effectue à droite — côté cible — et ψ résulte d'une composition de fonctions élémentaires d_j . La recherche bidirectionnelle s'arrête dès qu'il existe A_i et B_j tels que $\varphi(A_i) \sqsupseteq \psi(B_j)$.

article.

4 Discussion

Dans cet article, nous avons présenté une approche du raisonnement à partir de cas qui s'appuie sur le processus de classification dans un système de représentation de connaissances par objets. Les deux opérations principales qui constituent l'essentiel de la procédure d'inférence d'un système de RÀPC, la remémoration et l'adaptation, sont décrites en détail, et les rapports avec le raisonnement par classification sont explicités. Un des apports principaux de cette approche réside dans les notions de classifications dure et élastique, et dans celle de chemin de similarité. Ce travail se démarque des approches où la recherche d'un cas similaire s'appuie explicitement sur une mesure de similarité : les opérations de remémoration et d'adaptation sont combinées dans la notion de chemin de similarité, qui devient la base de travail pour la construction d'une solution du problème cible considéré. La faisabilité de cette approche s'est matérialisée dans la conception du système RESYN/RÀPC, qui est une extension du système de planification de synthèse en chimie organique RESYN [Vismara, 1995], RESYN/RÀPC étant chargé de la gestion des plans de synthèse sur la base du RÀPC. Il n'est pas directement question de RESYN/RÀPC dans cet article (voir [Lieber et Napoli, 1996], [Lieber et Napoli, 1997] et [Lieber, 1997]), mais ce sont plutôt les idées sur lesquelles repose ce système qui sont mises en valeur, et en particulier comment les classifications dure et élastique permettent de rechercher la solution d'un problème de synthèse pouvant se formaliser comme un problème de requête dans une RCO.

L'analyse qui précède constitue un travail formel de mise en forme et d'explication, devant être poursuivie dans plusieurs directions. Ainsi, la forme des index, des conjonctions d'attributs - valeurs, peut être généralisée, en y incluant des disjonctions et des négations,

comme c'est le cas pour les concepts de la logique de descriptions \mathcal{ALCNR} [Buchheit et al., 1993], ce qui permettrait de considérer des requêtes et donc des problèmes de recherche beaucoup plus complexes. Par ailleurs, il reste également à expliciter la nature du processus d'indexation, les relations qui existent entre un énoncé de problème et son index, ainsi que les conditions sous lesquelles un processus d'indexation automatique est envisageable.

Il existe un certain nombre de travaux qui sont en rapport avec le travail décrit ici, à commencer par [Koehler, 1996]. Dans [Koehler, 1996], la planification à partir de cas est présentée dans le contexte des logiques de descriptions et est analysée d'un point de vue formel logique : les cas sont des plans représentés par des formules dans une logique temporelle, et les index de ces cas sont représentés par des concepts dans une logique de descriptions ; il y a donc un changement de formalisme de représentation, ce qui n'est pas le cas dans notre approche. L'article montre ensuite comment est réalisée l'adéquation entre les deux formalismes, et comment sont traitées les opérations de remémoration, d'adaptation et de mémorisation. Le lien avec notre approche réside surtout dans l'exploitation commune d'une hiérarchie d'index par l'intermédiaire du processus de classification, comme mécanisme de base pour la remémoration.

Un autre travail en relation avec le nôtre est présenté dans [Plaza, 1995], qui utilise le formalisme des *feature terms* : les cas sont représentés par des formules conceptuelles — ou termes — munies de caractéristiques (pratiquement, des conjonctions d'attributs comme les classes présentées plus haut) et qui sont organisées en une hiérarchie grâce à une relation de subsumption spéciale. La recherche d'un cas similaire s'appuie sur la notion d'*antiunification*, qui matérialise ce que deux termes comparés partagent. En utilisant l'antiunification, il est possible de construire une description de la similitude structurelle existant

entre deux termes, et donc de classer les cas d'une base en fonction de leur similitude par rapport à un terme donné. Ce travail s'appuie plutôt sur la mise en évidence des différences entre les cas sources et le cas cible pour favoriser l'opération de remémoration, sans que l'adaptation soit considérée comme la préoccupation principale, alors que pour notre part, nous mettons en valeur les opérations d'adaptation élémentaires permettant de passer d'un cas source à un cas cible.

La notion d'antiunification peut être rapprochée des notions de *plus petit subsumant commun* ou PPSC — introduite sous le nom de LCS dans [Cohen *et al.*, 1992] et reprise dans [Coupey *et al.*, 1998] et [Salotti et Ventos, 1997] — ainsi que l'opération d'*intersection* introduite dans [Beck, 1991]. Comme l'antiunification, les opérations de PPSC et d'intersection essaient de construire une description minimale généralisant deux descriptions données, afin d'évaluer les similarités et dissimilarités entre ces deux descriptions. Ainsi, dans [Coupey *et al.*, 1998] et [Salotti et Ventos, 1997], l'opération de remémoration s'appuie sur l'opération de PPSC et le processus de classification. L'univers de ces travaux est tout à fait original, car il est constitué de la logique de descriptions $\mathcal{AL}_{\delta\epsilon}$, qui intègre des constructeurs capables de prendre en compte des valeurs par défauts et des exceptions.

Nous terminerons cette discussion en évoquant deux autres problèmes importants :

(i) la représentation des cas : un cas est un élément de connaissance qui peut être accompagné d'un mode d'emploi et d'explications (ces dernières étant plutôt ignorées en général), et l'utilisation d'une RCO favorise une représentation homogène des cas, de leurs index, et des connaissances éventuelles les accompagnant,

(ii) l'implantation et l'évaluation des systèmes à base de cas : les courbes temporelles mesurant l'efficacité en termes de temps de calcul ne sont pas toujours totalement adéquates pour donner une bonne évaluation ; il reste encore à imaginer une véritable évaluation qualitative globale par rapport à certaines propriétés, comme par exemple les coûts qualitatifs de la remémoration et de l'adaptation, qui sont des points qui doivent être examinés beaucoup plus en détail.

5 Conclusion

Dans cet article, nous avons présenté les notions de classifications dure et élastique, et celle de chemin de similarité, qui permettent de concevoir un système de raisonnement à partir de cas à l'aide d'un système de représentation de connaissances par objets. Nous avons montré que les liens entre raisonnement par classification et raisonnement à partir de cas sont très

étroits dès lors que les cas et les index associés peuvent être représentés par des objets qui sont organisés en hiérarchie. Par ailleurs, la notion de généralité entre problèmes permet de fournir une base explicite pour organiser les problèmes de façon hiérarchique. Ce travail d'analyse doit permettre de mieux comprendre les liens entre raisonnement à partir de cas et raisonnement par classification, ainsi que la notion de remémoration guidée par l'adaptabilité. Ce travail possède de nombreuses perspectives de recherches qui vont nous occuper — ou nous occupent déjà — comme la prise en compte d'index et de problèmes plus complexes (introduction de la négation et de la disjonction dans le formalisme de représentation) et le traitement de la phase d'apprentissage associée au cycle du raisonnement à partir de cas.

Remerciements

Les auteurs tiennent à remercier l'Institut de Recherches Servier pour l'aide financière qu'il a accordée au premier auteur, ainsi que les lecteurs anonymes, dont les remarques et les suggestions ont contribué à l'amélioration de cet article.

Les auteurs font (également) partie du GDR 1093 du CNRS « Traitement informatique de la connaissance en chimie organique », et c'est dans ce cadre que sont développées les recherches présentées ici.

Références

- [Beck, 1991] H.W. Beck. Language Acquisition from Cases. Dans *Proceedings of the DARPA Case-Based Reasoning Workshop, Washington, D.C.*, pages 159–169, 1991.
- [Borgida et McGuinness, 1996] A. Borgida et D.L. McGuinness. Asking Queries about Frames. Dans *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, Cambridge, Massachusetts, pages 340–349, 1996.
- [Buchheit *et al.*, 1993] M. Buchheit, F.M. Donini et A. Schaerf. Decidable Reasoning in Terminological Knowledge Representation Systems. *Journal of Artificial Intelligence Research*, 1:109–138, 1993.
- [Carbonell, 1983] J.G. Carbonell. Learning by analogy: Formulating and generalizing plans from past experiences. Dans *Machine Learning, an Artificial Intelligence Approach, Volume I*, éditeurs R.S. Michalski, J.G. Carbonell et T.M. Mitchell, pages 137–162. Morgan Kaufmann Publishers, Inc., Los Altos, California, 1983.
- [Chouvet *et al.*, 1996] M.-P. Chouvet, F. Le Ber, J. Lieber, L. Mangelinck, A. Napoli et A. Simon. Analyse des besoins en représentation et raisonne-

- ment dans une représentation à objets — L'exemple de Y3. Dans *Actes du colloque "Langages et Modèles à Objets" (LMO'96)*, Leysin, Suisse, éditeur Y. Dennebouy, pages 150–169. École Polytechnique Fédérale de Lausanne (EPFL-LBD), 1996.
- [Cohen et al., 1992] W. Cohen, A. Borgida et H. Hirsh. Computing Least Common Subsumers in Description Logics. Dans *Proceedings of AAAI'92, San Jose, California*, pages 754–760, 1992.
- [Coupey et al., 1998] P. Coupey, C. Fouqueré et S. Salotti. Formalizing partial matching and similarity in CBR with a description logic. *Applied Artificial Intelligence Journal*, 12(1):71–112, 1998.
- [Koehler, 1996] J. Koehler. Planning from Second Principles. *Artificial Intelligence*, 87:145–186, 1996.
- [Lieber et Napoli, 1996] J. Lieber et A. Napoli. Using Classification in Case-Based Planning. Dans *European Conference on Artificial Intelligence (ECAI'96), Budapest, Hungary*, éditeur W. Wahlgster, pages 132–136. John Wiley & Sons Ltd, Chichester, 1996.
- [Lieber et Napoli, 1997] J. Lieber et A. Napoli. Planification à partir de cas et classification. Dans *Actes des Journées Ingénierie des Connaissances et Apprentissage Automatique, Roscoff*, éditeur J. Charlet, pages 617–628. INRIA Rennes, 1997.
- [Lieber, 1997] J. Lieber. *Raisonnement à partir de cas et classification hiérarchique. Application à la planification de synthèse en chimie organique*. Thèse, Université Henri Poincaré, Nancy, 1997.
- [Mignot, 1997] H. Mignot. *Sensibilité au contexte lors de l'évaluation de la similarité dans une architecture ouverte pour le raisonnement à partir de cas*. Thèse, Université de Paris-Sud, Centre d'Orsay, 1997.
- [Mugnier et Chein, 1996] M.L. Mugnier et M. Chein. Représenter des connaissances et raisonner avec des graphes. *Revue d'intelligence artificielle*, 10(1):7–56, 1996.
- [Napoli et Laurenço, 1993] A. Napoli et C. Laurenço. Représentations à objets et classification – Conception d'un système d'aide à la planification de synthèses organiques. *Revue d'intelligence artificielle*, 7(2):175–221, 1993.
- [Napoli, 1994] A. Napoli. Catégorisation, raisonnement par classification et raisonnement à partir de cas. Dans *Journées Acquisition – Validation – Apprentissage (JAVA'94), Strasbourg*, pages E1–E14, 1994.
- [Napoli, 1995] A. Napoli. Objects, Classes, Specialization and Subsumption. Dans *Proceedings of the 1995 International Workshop on Description Logics, Università di Roma (Technical Report 07.95)*, éditeurs A. Borgida, M. Lenzerini, D. Nardi et B. Nebel, pages 52–55, 1995.
- [Nebel, 1990] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*. Lecture Notes in Computer Science 422. Springer-Verlag, Berlin, 1990.
- [Plaza, 1995] E. Plaza. Cases as terms: A feature term approach to the structured representation of cases. Dans *Case-Based Reasoning. Research and Development, Proceedings of the First International Conference on Case-Based Reasoning (ICCBR'95), Seimbra, Portugal*, éditeurs M. Veloso et A. Aamodt, Lecture Notes in Artificial Intelligence 1010, pages 265–275. Springer Verlag, Berlin, 1995.
- [Rifqi, 1996] M. Rifqi. *Mesures de comparaison, typicité et classification d'objets flous : théorie et pratique*. Thèse, Université Pierre et Marie Curie (Paris 6), 1996.
- [Rougegrez-Loriette, 1994] S. Rougegrez-Loriette. *Prédiction de processus à partir de comportements observés : le système REBECAS*. Thèse, Université Pierre et Marie Curie (Paris 6), 1994.
- [Salotti et Ventos, 1997] S. Salotti et V. Ventos. Étude et formalisation d'un système de Raisonnement à Partir de Cas dans une Logique de Descriptions. Dans *Actes des Journées Ingénierie des Connaissances et Apprentissage Automatique, Roscoff*, éditeur J. Charlet, pages 629–642. INRIA Rennes, 1997.
- [Smyth et Keane, 1994] B. Smyth et M.T. Keane. Retrieving Adaptable Cases: The Role of Adaptation Knowledge in Case Retrieval. Dans *Topics in Case-Based Reasoning – First European Workshop (EWCBR'93), Kaiserslautern*, éditeurs S. Wess, K.-D. Althoff et M.M. Richter, Lecture Notes in Artificial Intelligence 837, pages 209–220, Berlin, 1994. Springer-Verlag.
- [Smyth, 1996] B. Smyth. *Case-Based Design*. PhD thesis, Trinity College, University of Dublin, Ireland, 1996.
- [Stefik, 1995] M. Stefik. *Introduction to Knowledge Systems*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1995.
- [Veloso, 1994] M. M. Veloso. *Planning and Learning by Analogical Reasoning*. Lecture Notes in Artificial Intelligence 886. Springer, Berlin, 1994.
- [Vismara, 1995] P. Vismara. *Reconnaissance et représentation d'éléments structuraux pour la description d'objets complexes. Application à l'élaboration de stratégies de synthèse en chimie organique*. Thèse, Université des Sciences et Techniques du Languedoc, Montpellier, 1995.